Workbook Object

```
Application
  └ Workbooks [Workbook]
       └ Multiple Objects
```
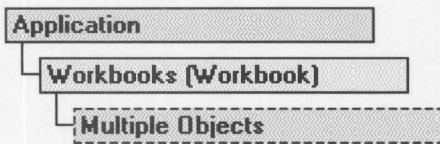
Represents a Microsoft Excel workbook. The **Workbook** object is a member of the **Workbooks** collection. The **Workbooks** collection contains all the **Workbook** objects currently open in Microsoft Excel.

Using the Workbook Object

The following properties for returning a **Workbook** object are described in this section:

- **Workbooks** property

- **ActiveWorkbook** property

- **ThisWorkbook** property

Workbooks Property

Use **Workbooks**(*index*), where *index* is the workbook name or index number, to return a single **Workbook** object. The following example activates workbook one.

```
Workbooks(1).Activate
```

The index number denotes the order in which the workbooks were opened or created. `Workbooks(1)` is the first workbook created, and `Workbooks(Workbooks.Count)` is the last one created. Activating a workbook doesn't change its index number. All workbooks are included in the index count, even if they're hidden.

The **Name** property returns the workbook name. You cannot set the name by using this property; if you need to change the name, use the **SaveAs** method to save the workbook under a different name. The following example activates Sheet1 in the workbook named "Cogs.xls" (the workbook must already be open in Microsoft Excel).

```
Workbooks("cogs.xls").Worksheets("sheet1").Activate
```

ActiveWorkbook Property

The **ActiveWorkbook** property returns the workbook that's currently active. The following example sets the name of the author for the active workbook.

```
ActiveWorkbook.Author = "Jean Selva"
```

ThisWorkbook Property

The **ThisWorkbook** property returns the workbook where the Visual Basic code is running. In most cases, this is the same as the active workbook. However, if the Visual Basic code is part of an add-in, the **ThisWorkbook** property won't return the active workbook. In this case, the active workbook is the workbook calling the add-in, whereas the **ThisWorkbook** property returns the add-in workbook.

If you'll be creating an add-in from your Visual Basic code, you should use the **ThisWorkbook** property to qualify any statement that must be run on the workbook you compile into the add-in.